

IN THE CLAIMS:

- A4
1. (Original) A method of handling exceptions in a device having predication, comprising:
 - determining if an exception is pending based on values of a predicate register pair; and
 - handling the exception when it is determined that an exception is pending.
 2. (Original) The method of claim 1, wherein determining if an exception is pending includes determining if a value of a first predicate register is true and a second predicate register is false.
 3. (Original) The method of claim 1, wherein handling the exception includes determining if an address of an instruction within a method that threw the exception is in a try block, and if the address of the instruction is not in the try block, invoking a return associated with the method.
 4. (Original) The method of claim 3, wherein if the exception is in the try block, using an associated exception handler for the method.
 5. (Original) The method of claim 1, wherein the device has an IA64 architecture.
 6. (Currently amended) The method of claim 1, wherein handling the exception includes determining if [[the]] an instruction in a method that threw the exception is in a try block and invoking a snippet associated with the method.
 7. (Original) The method of claim 6, wherein the snippet invokes a lookup handler for determining if the exception is within a try block of the method.
 8. (Original) The method of claim 7, wherein the lookup handler determines if the exception is within the try block of the method by searching an exception table associated

with the method and determining if an address of the instruction is within the exception table.

A4

9. (Original) An apparatus for handling exceptions in a device having predication, comprising:

means for determining if an exception is pending based on values of a predicate register pair; and

means for handling the exception when it is determined that an exception is pending.

10. (Original) The apparatus of claim 9, wherein the means for determining if an exception is pending determines if a value of a first predicate register is true and a second predicate register is false.

11. (Original) The apparatus of claim 9, wherein the means for handling the exception determines if an address of an instruction within a method that threw the exception is in a try block, and if the address of the instruction is not in the try block, invokes a return associated with the method.

12. (Original) The apparatus of claim 11, wherein if the exception is in the try block, the means for handling uses an associated exception handler for the method.

13. (Original) The apparatus of claim 9, wherein the apparatus has an IA64 architecture.

14. (Currently amended) The apparatus of claim 9, wherein the means for handling the exception determines if [[the]] an instruction in a method that threw the exception is in a try block and invokes a snippet associated with the method.

15. (Original) The apparatus of claim 14, wherein the snippet invokes a lookup handler for determining if the exception is within a try block of the method.

16. (Original) The apparatus of claim 15, wherein the lookup handler determines if the exception is within the try block of the method by searching an exception table associated with the method and determining if an address of the instruction is within the exception table.

17. (Original) A computer program product in a computer readable medium for handling exceptions in a device having predication, comprising:

first instructions for determining if an exception is pending based on values of a predicate register pair; and

second instructions for handling the exception when it is determined that an exception is pending.

18. (Original) The computer program product of claim 17, wherein the first instructions for determining if an exception is pending includes instructions for determining if a value of a first predicate register is true and a second predicate register is false.

19. (Original) The computer program product of claim 17, wherein the second instructions for handling the exception includes instructions for determining if an address of an instruction within a method that threw the exception is in a try block, and instructions for invoking a return associated with the method if the address of the instruction is not in the try block.

20. (Original) The computer program product of claim 19, wherein the second instructions further include instructions for using an associated exception handler for the method if the exception is in the try block.

21. (Original) The computer program product of claim 17, wherein the device has an IA64 architecture.

A4
22. (Currently amended) The computer program product of claim 17, wherein the second instructions for handling the exception includes instructions for determining if [[the]] an instruction in a method that threw the exception is in a try block and instructions for invoking a snippet associated with the method.

23. (Original) The computer program product of claim 22, wherein the snippet invokes a lookup handler for determining if the exception is within a try block of the method.

24. (Original) The computer program product of claim 23, wherein the lookup handler determines if the exception is within the try block of the method by searching an exception table associated with the method and determining if an address of the instruction is within the exception table.